AMENDMENTS TO THE SPECIFICATION:

Please amend the specification as follows:

Please replace paragraphs [001]-[004], [033], and [056]-[060] with the following paragraphs:

[001] Related U. S. Patent Application Serial No. 10/035,747, filed on even date herewith in the name of Guy L. Steele, Jr. and entitled "Floating Point System That Represents Status Flag Information Within A Floating Point Operand," assigned to the assignee of the present application, is hereby incorporated by reference.

[002] Related U. S. Patent Application Serial No. 10/035,595, filed on even date herewith in the name of Guy L. Steele, Jr. and entitled "Floating Point Adder with Embedded Status Information," assigned to the assignee of the present application, is also hereby incorporated by reference.

[003] Related U. S. Patent Application Serial No. 10/035,580, filed on even date herewith in the name of Guy L. Steele, Jr. and entitled "Floating Point Multiplier with Embedded Status Information," assigned to the assignee of the present application, is also hereby incorporated by reference.

[004] Related U. S. Patent Application Serial No. 10/035,647, filed on even date herewith in the name of Guy L. Steele, Jr. and entitled "Floating Point Divider with Embedded Status Information," assigned to the assignee of the present application, is also hereby incorporated by reference.

[033] U.S. Patent Application Serial No. 10/035,584, filed on even date herewith in the name of Guy L. Steele Jr. and entitled "Floating Point Unit In Which Floating Point Status Information Is Encoded In Floating Point Representations," describes a floating point unit in which floating point status information is encoded in the representations of the results generated thereby. By encoding the floating point status information relating to a floating point operation in the result that is generated for the operation, the implicit serialization required by maintaining the floating point status information separate and apart therefrom can be obviated. The floating point unit includes a plurality of functional units, including an adder unit, a multiplier unit, a divider unit, a square root unit, a maximum/minimum unit, a comparator unit and a tester unit, all of which operate under control of functional unit control signals provided by a control unit. It may also include features consistent with the principles of the present invention to provide better support for interval arithmetic.

[056] Systems and methods consistent with the principles of the present invention provide support for floating-point arithmetic in the manner of IEEE 754 and in the manner described in related, incorporated-by-reference U. S. Patent Application Serial No. 10/035,747, filed on even date herewith in the name of Guy L. Steele, Jr. entitled "Floating Point System That Represents Status Flag Information Within A Floating Point Operand," assigned to the assignee of the present application. Embodiments consistent with the principles of the present invention define results for arithmetic operations with the rounding modes "round toward plus infinity" and "round toward minus infinity" such that a "NaN" is not produced as a result if none of the inputs is a NaN. Eliminating NaN as a result improves interval arithmetic calculations because

valid intervals are produced instead of malformed intervals with NaN as an endpoint. Further, substituting a non-NaN result for the conventional NaN result allows for efficient processing because execution time and code space are not wasted on special routines otherwise used to handle a conventional NaN result. Systems and methods consistent with the principles of the present invention may comprise software to be used for performing addition, subtraction, multiplication, and division of intervals on a computer system. Software that supports the improved definition of floating-point addition, subtraction, multiplication, and division, performs interval computations that produce more precise and reliable results than computations performed by existing systems.

[057] One embodiment consistent with the principles of the invention supports floating-point arithmetic as defined by related U. S. Patent Application Serial No. 10/035,747, filed on even date herewith in the name of Guy L. Steele, Jr. entitled "Floating Point System That Represents Status Flag Information Within A Floating Point Operand," assigned to the assignee of the present application. In this embodiment, the floating point operands have status information encoded within the operand itself. Additionally, the floating point arithmetic includes computations involving positive and negative overflow and underflow values (+OV, -OV, +UN, and -UN). Another embodiment consistent with the principles of the invention supports IEEE 754-compliant arithmetic, except for the changes in the handling of zero and infinity operands as described herein.

[058] Related U. S. Patent Application Serial No. 10/035,595, filed on even date herewith in the name of Guy L. Steele, Jr. and entitled "Floating Point Adder with Embedded Status Information," assigned to the assignee of the present application,

describes improved add and subtract instructions. In brief summary, subtraction is defined to behave as (a - b) = (a + (-b)), where unary "-" simply flips the sign bit of its operand. As shown in FIG. 2, an embodiment consistent with the principles of the present invention improves addition operations 205 and subtraction operations 207 by providing that for "round toward plus infinity" the result of adding infinities of opposite signs is +Infinity, substituted for NaN as specified by IEEE Std. 754. Similarly, for "round toward minus infinity" addition 206 and subtraction 208 operations, the result of adding infinities of opposite signs is -Infinity, instead of NaN. As a result, the standard formulae for interval addition and subtraction:

$$[a,\ b] + [c, d] = [a\ +\mathrm{DOWN}\ c,\ b\ +\mathrm{UP}\ d]$$
$$[a,\ b] - [c, d] = [a\ -\mathrm{DOWN}\ d,\ b\ -\mathrm{UP}\ c]$$

produce improved results in certain situations where IEEE 754 would produce a malformed interval (e.g., an interval with a NaN) or would necessitate the use of a more complicated formula to forestall the production of a NaN result.

[059] Related U. S. Patent Application Serial No. 10/035,580, filed on even date herewith in the name of Guy L. Steele, Jr. and entitled "Floating Point Multiplier with Embedded Status Information," assigned to the assignee of the present application, describes an improved multiplication instruction. An embodiment consistent with the principles of the present invention improves multiplication operations by providing that a multiplication with "round toward plus infinity" 210 produces -0 as the result of multiplying an infinity and a zero of opposite signs, substituting for NaN as specified by IEEE 754. Similarly, the result of multiplying an infinity and a zero of like signs is forced to +Infinity, instead of resulting in a NaN. Also, for "round toward minus infinity"

multiplication 215, the result of multiplying an infinity and a zero of opposite signs is -Infinity, not NaN, and the result of multiplying an Infinity and a zero of like signs is +0, not NaN. Using such an implementation, the standard formula for interval multiplication:

$$[a, b] * [c, d] = [\min(a *\text{DOWN } c, a *\text{DOWN } d, b *\text{DOWN } c, b *\text{DOWN } d), \max(a *\text{UP } c, a *\text{UP } d, b *\text{UP } c, b *\text{UP } d)]$$

produces improved results in certain situations where IEEE 754 would otherwise produce a malformed interval (e.g., an interval containing a NaN) or would necessitate the use of a more complicated formula to forestall the production of a NaN result.

[060] Related U. S. Patent Application Serial No. 10/035,647, filed on even date herewith in the name of Guy L. Steele, Jr. and entitled "Floating Point Divider with Embedded Status Information," assigned to the assignee of the present application, describes an improved division instruction. An embodiment consistent with the principles of the present invention improves divide operations by providing that for "round toward plus infinity" division 220, the result of dividing two infinities or two zeros of opposite signs is -0, substituted for NaN as specified by IEEE 754. Additionally, the result of dividing two infinities or two zeros of like signs is +Infinity, instead of NaN. Similarly, for "round toward minus infinity" division 225, the result of dividing two infinities or two zeros of opposite signs is -Infinity, not NaN, and the result of dividing two infinities or two zeros of like sign is +0, substituted for NaN. Using such an improvement, the standard formula for interval division:

$$[a, b] / [c, d] = \text{if } (c > 0 \text{ or } d < 0) \text{ then } [\min(a /\text{DOWN } c, a /\text{DOWN } d, b *\text{DOWN } c, b *\text{DOWN } d), \max(a /\text{UP } c, a /\text{UP } d, b /\text{UP } c, b /\text{UP } d)] \text{ else } [-\text{Infinity}, +\text{Infinity}]$$

produces improved results in certain situations where IEEE 754 would produce a malformed interval (e.g., an interval containing a NaN) or would necessitate the use of a more complicated formula to forestall the production of a NaN result.